Ejercicios de regresión

Asignatura	Métodos Avanzados en Estadística
Correo	gloria.valle@estudiante.uam.es
Día	@December 9, 2021
Estudiante	Gloria del Valle Cano
Tema	Tema 3
Tipo	Ejercicios

Ejercicio 1

Los datos del fichero *Datos-geyser.txt* corresponden al día de la observación (primera columna), el tiempo medido en minutos (segunda columna Y) y el tiempo hasta la siguiente erupción (tercera columna X) del geyser *Old Faithful* en el parque norteamericano de *Yellowstone*.

(a) Representa gráficamente los datos, junto con el estimador de Nadaraya-Watson de la función de regresión de Y sobre X.
(b) Representa gráficamente los datos, junto con el estimador localmente lineal de la función de regresión de Y sobre X.

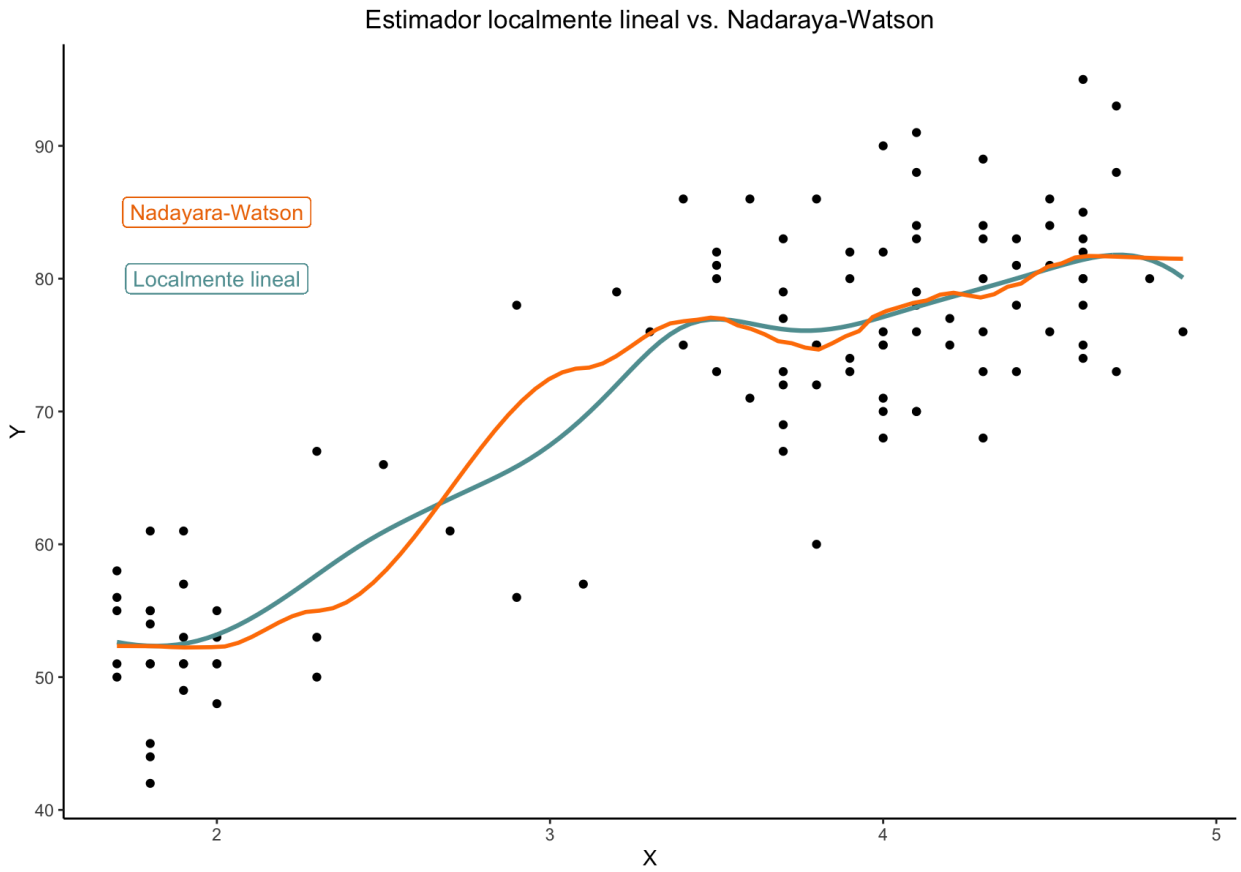
```
library(ggplot2)
library(dplyr)
library(KernSmooth)

# lectura del fichero

df <- read.table("https://matematicas.uam.es/~joser.berrendero/datos/Datos-geyser.txt",
                 header=FALSE, skip=1, sep=' ', col.names=c(' ', 'D', 'Y', 'X'),
                 colClasses=c('numeric', 'numeric', 'numeric', 'numeric'))
# guardado en dataframe
df <- df[,c('D', 'Y', 'X')]

# ajuste localmente lineal
ajuste <- with(df, locpoly(X, Y, degree=1, bandwidth=dpill(X, Y), gridsize=107))
ajuste

# plot de ambas
df %>%
  mutate(curva=ajuste$y) %>%
  ggplot() +
  ggtitle('Estimador localmente lineal vs. Nadaraya-Watson') +
  theme(plot.title = element_text(hjust = 0.5)) +
  geom_point(aes(X, Y)) +
  geom_line(aes(ajuste$x, curva), color='cadetblue', size=1.1) +
  geom_label(aes(x=2, y=80, label='Localmente lineal'), color='cadetblue') +
  geom_smooth(formula = y ~ x, mapping=aes(X, Y), method='loess', se=FALSE, span=0.25, method.args=list(degree=0), col='darkorange1') +
  geom_label(aes(x=2, y=85, label='Nadayara-Watson'), color='darkorange2')+
  theme(axis.line = element_line(colour = "black"),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.border = element_blank(),
        panel.background = element_blank())
```



Ejercicio 4

Se considera el siguiente modelo de regresión lineal múltiple:

$$Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \epsilon_i, \qquad \epsilon_i \equiv \mathcal{N}(0, \sigma^2), \;\; i = 1, ..., n. \qquad (1)$$

Se dispone de $n = 20$ observaciones con las que se ajustan todos los posibles submodelos del modelo, obteniéndose para cada uno de ellos las siguientes sumas de cuadrados de los residuos (todos los submodelos incluyen un término independiente).

Variables incluidas en el modelo	SCR
Sólo término independiente	42644.00
x_1	8352.28
x_2	36253.69
x_3	36606.19
x_1 y x_2	7713.13
x_1 y x_3	762.55
x_2 y x_3	**32700.17**
x_1, x_2 y x_3	761.41

Ejemplo en negrita: Para el modelo ajustado $\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_2 x_{i2} + \hat{\beta}_3 x_{i3}$ la suma de los cuadrados de los residuos es 32700.17.

(a) Calcula la tabla de análisis de la varianza para el modelo (1) y contrasta a nivel $\alpha = 0.005$ la hipótesis nula $H_0 : \beta_1 = \beta_2 = \beta_3 = 0$.

La tabla de análisis se obtiene con la suma de cuadrados explicada (SCE), lo cual podemos deducir tomando que para un modelo $Y_i = \beta_0 + \epsilon_i$, el valor de β_0 que minimiza el MSE entre la predicción y los valores reales es $\hat{\beta}_0 = \bar{Y}$. Teniendo que SCE_0 es la suma de cuadrados de residuos del modelo ajustado bajo la hipótesis nula, en este caso, los residuos del término independiente y el SCR:

$$SCR_0 - SCR = \sum(\underbrace{\hat{\beta}_0}_{\bar{Y}_i} - Y_i)^2 - \sum(\hat{Y}_i - Y_i)^2 = SCT - SCR = SCE$$

Lo cual nos lleva a la obtención del SCE.

```
# guardado de los datos
scrs <- c(42644, 8352.28, 36253.69, 36606.19, 7713.13, 762.55, 32700.17, 761.41)
vars <- c('Indep.', 'x1', 'x2', 'x3', 'x1 y x2', 'x1 y x3', 'x2 y x3', 'x1, x2 y x3')
datos <- data.frame(scrs=scrs, vars=vars)
# datos completos
n <- 20 # numero de observaciones
p <- 3 # numero de variables
scr <- tail(scrs, n=1)
sce <- scrs[1]-scr
df <- c(p, n-p-1)
sum_sq <- c(sce, scr)
mean_sq <- sum_sq/df
f_value <- c(mean_sq[1]/mean_sq[2], NA)
pr <- c(pf(f_value[1], df1=p, df2=n-p-1, lower.tail=FALSE), NA)

anova <- data.frame("anova data"=c('scrs', 'Residuals'), 'Df'=df, 'Sum Sq'=sum_sq,
                    'Mean Sq'=mean_sq, 'F value'=f_value, 'Pr(>F)'=pr)
```

```
> anova
anova data Df   Sum Sq     Mean Sq  F value       Pr(>F)
1       scrs  3 41882.59 13960.86333 293.3686 3.421043e-14
2  Residuals 16   761.41    47.58812       NA           NA
```

Para contrastar H_0 se emplea el estadístico F, ya que la variación total coincide con la residual en el modelo que ajusta únicamente el término independiente.

$$F = \frac{\frac{SCE}{3}}{\frac{SCR}{16}} = 293.3686$$

```
> f_value
[1] 293.3686       NA
> pr
[1] 3.421043e-14           NA
```

Como para $\mathbb{P}(F_{3;16} > F) = 3.421043e-14$, lo que es mucho menor que $\alpha = 0.05$, rechazamos la hipótesis nula de que β_1, β_2 y β_3 sean 0 salvo β_0.

(b) En el modelo (1), contrasta a nivel $\alpha = 0.05$ las dos hipótesis nulas siguientes:

- $H_0 : \beta_2 = 0$
- $H_0 : \beta_1 = \beta_3 = 0$

Para contrastar ambas hipótesis, buscaremos primero el estadístico F en cada caso.

$$F = \frac{\frac{SCR_0 - SCR}{k}}{\frac{SCR}{n-p-1}}$$

Por aclarar, k es el número de filas de la matriz A de coeficientes del modelo que verifica $H_0 : A\beta = 0$.

- Para el caso $H_0 : \beta_2 = 0$, tenemos que $k = 1$.

```
k <- 1
f_value <- ((scrs[6]-scr)/k / (scr/(n-p-1)))
pr <- pf(f_value, df1=k, df2=n-p-1, lower.tail=FALSE)
```

```
> pr
[1] 0.8789337
```

Como el $p-$valor calculado es mayor que $\alpha = 0.05$ no se tiene evidencia suficiente como para rechazar la hipótesis nula.

- Para el caso $H_0 : \beta_1 = \beta_3 = 0$, tenemos que $k = 2$.

```
k <- 2
f_value <- ((scrs[3]-scr)/k / (scr/(n-p-1)))
pr <- pf(f_value, df1=k, df2=n-p-1, lower.tail=FALSE)
```

```
> pr
[1] 3.785566e-14
```

Como el $p-$valor calculado es mucho menor que $\alpha = 0.05$ se rechaza la hipótesis nula.

Ejercicio 6

Sean Y_1, Y_2 e Y_3 tres variables aleatorias independientes con distribución normal y varianza σ^2. Supongamos que μ es la media de Y_1, λ es la media de Y_2 y $\lambda + \mu$ es la media de Y_3, donde $\lambda, \mu \in \mathbb{R}$.

(a) Demuestra que el vector $Y = (Y1, Y2, Y3)'$ verifica el modelo de regresión múltiple $Y = X\beta + \epsilon$. Para ello, determina la matriz de diseño X, el vector de parámetros β y la distribución de las variables de error ϵ.

El modelo de regresión se puede expresar como:

$$\underbrace{\begin{pmatrix} Y_1 \\ Y_2 \\ Y_3 \end{pmatrix}}_{Y} = X \underbrace{\begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix}}_{\beta} + \underbrace{\begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \end{pmatrix}}_{\epsilon}$$

Teniendo en cuenta que $Y = (\mu, \lambda, \lambda + \mu)$ y que $\epsilon \sim \mathcal{N}_n(0, \sigma^2 \mathbb{I}_n) \iff Y \sim \mathcal{N}_n(X\beta, \sigma^2 \mathbb{I}_n)$ por la propiedad de transformaciones lineales de la normal multivariante.

$$\underbrace{\begin{pmatrix} Y_1 \\ Y_2 \\ Y_3 \end{pmatrix}}_{Y} = X \underbrace{\begin{pmatrix} \mu \\ \lambda \end{pmatrix}}_{\beta} + \underbrace{\begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \end{pmatrix}}_{\epsilon}$$

Por tanto la matriz de diseño X es igual a:

$$X = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix}$$

(b) Calcula los estimadores de máxima verosimilitud (equivalentemente, de mínimos cuadrados) de λ y μ.

Tomando la expresión de mínimos cuadrados de β que se obtiene como solucion de las ecuaciones normales, $\hat{\beta} = (X'X)^{-1}X'Y$ y que la matriz $X'X$ es simétrica:

$$\hat{\beta} \sim \mathcal{N}_2((X'X)X'X\beta, \sigma^2 X((X'X)^{-1})(X'X)^{-1}X')$$

Lo que es prácticamente:

$$\hat{\beta} \sim \mathcal{N}_2(\beta, \sigma^2 (X'X)^{-1})$$

Teniendo esto, obtenemos $(X'X)^{-1}$:

$$X'X = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$$

La cual invertimos:

$$(X'X)^{-1} = \begin{pmatrix} 2/3 & -1/3 \\ -1/3 & 2/3 \end{pmatrix}$$

Finalmente:

$$\hat{\beta} = (X'X)^{-1}X'Y = \begin{pmatrix} 2/3 & -1/3 \\ -1/3 & 2/3 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} Y_1 \\ Y_2 \\ Y_3 \end{pmatrix} = \begin{pmatrix} 2/3Y_1 - 1/3Y_2 + 1/3Y_3 \\ -1/3Y_1 + 2/3Y_2 + 1/3Y_3 \end{pmatrix}$$

(c) Calcula la distribución del vector $(\hat{\lambda}, \hat{\mu})'$, formado por los estimadores calculados en el apartado anterior.

La distribución de $\hat{\beta}$ es (según la explicación del apartado anterior):

$$\begin{pmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \end{pmatrix} \sim \mathcal{N}_2 \left(\begin{pmatrix} \mu \\ \lambda \end{pmatrix}, \sigma^2 \begin{pmatrix} 2/3 & -1/3 \\ -1/3 & 2/3 \end{pmatrix} \right)$$

Ejercicio 10

Los datos fuel2001 del fichero $combustible.RData$ (véase transparencias de clase) corresponden al consumo de combustible (y otras variables relacionadas) en los estados de EE.UU. Se desea explicar la variable $FuelC$ en función del resto de la información.

(a) Representa en un plano las dos primeras componentes principales de estos datos estandarizados (consulta la ayuda de $prcomp$). ¿Son suficientes estas dos componentes para explicar un alto porcentaje de la varianza?

```
load(url('http://verso.mat.uam.es/~joser.berrendero/datos/combustible.RData'))
fuel2001
```

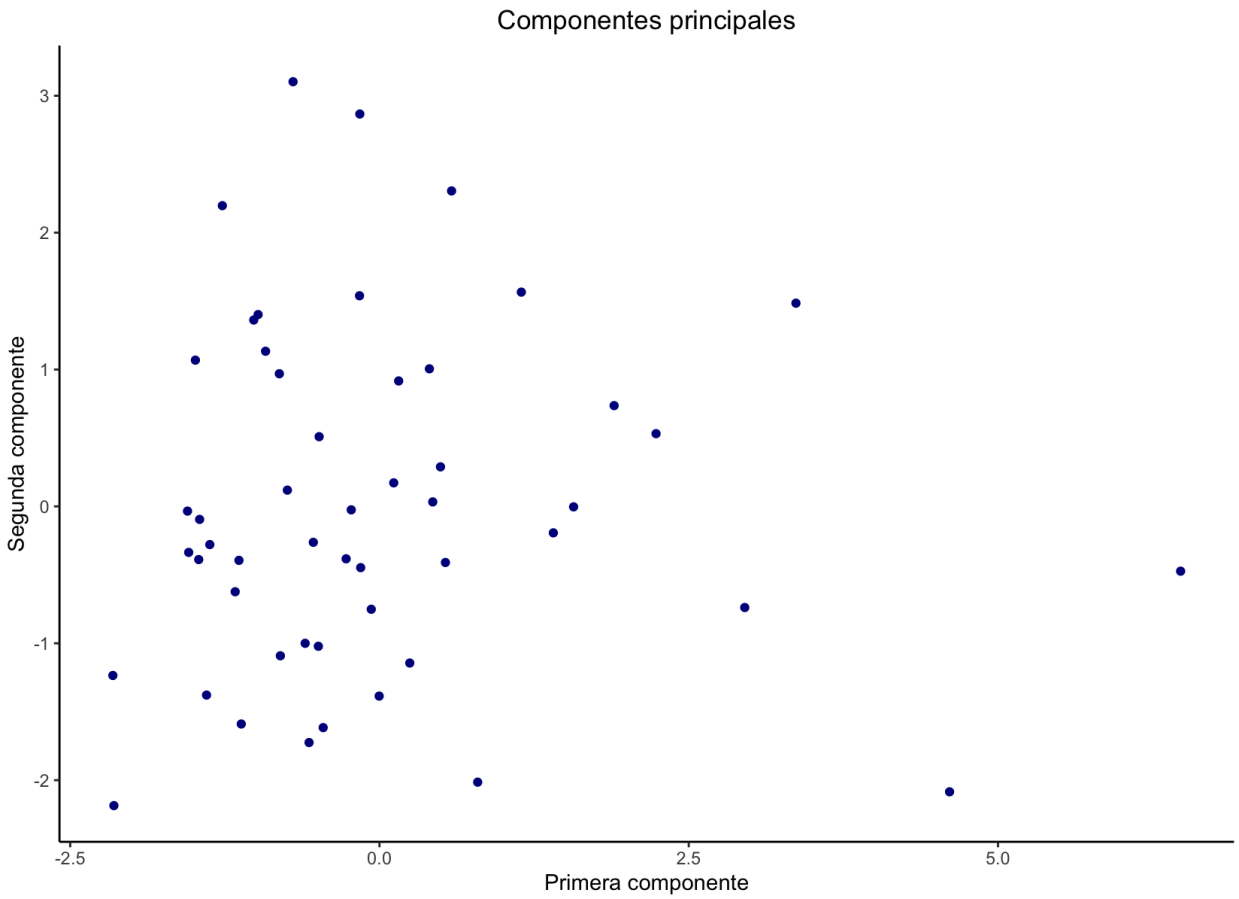
```
> head(fuel2001,10)
    Drivers    FuelC Income  Miles      MPC      Pop  Tax
AL  3559897  2382507  23471  94440 12737.00  3451586 18.0
AK   472211   235400  30064  13628  7639.16   457728  8.0
AZ  3550367  2428430  25578  55245  9411.55  3907526 18.0
AR  1961883  1358174  22257  98132 11268.40  2072622 21.7
CA 21623793 14691753  32275 168771  8923.89 25599275 18.0
CO  3287922  2048664  32949  85854  9722.73  3322455 22.0
CT  2650374  1458279  40640  20910  9021.35  2651452 25.0
DE   564099   382043  31255   5814 10891.30   610269 23.0
DC   328094   148769  37383   1534  6555.94   468575 20.0
FL 12743403  7471117  28145 117299  9531.23 12741821 13.6
```

Como nuestro objetivo es predecir la variable $FuelC$, la extraemos del conjunto para hacer PCA. Utilizamos la función $prcomp$ sugerida para dicho cálculo.

```
# Excluimos la variable a predecir y creamos un df con las dos componentes
p_comp <- prcomp(fuel2001[,-c(2)], scale=TRUE)
data <- data.frame(c1=p_comp$x[ ,1], c2=p_comp$x[ ,2])

# Plot de las componentes principales
ggplot(data, aes(x=c1, y=c2)) +
  geom_point(colour='darkblue') +
  ggtitle('Componentes principales') +
  xlab('Primera componente') +
  ylab('Segunda componente') +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme(axis.line = element_line(colour = "black"),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.border = element_blank(),
        panel.background = element_blank())

# Cálculo de la varianza acumulada
var_acumulada <- cumsum(p_comp$sdev^2)/sum(p_comp$sdev^2)
```



Para las dos primeras componentes se observa que el porcentaje es el siguiente:

```
> var_acumulada[2]
[1] 0.7190314
```

A pesar de que pueda parecer alto, no se alcanza al 80%, por lo que se considera que con dos variables no es muy alto.

(b) Ajusta el modelo completo con todas las variables. En este modelo completo, contrasta la hipótesis nula de que los coeficientes de las variables $Income$, MPC y Tax son simultáneamente iguales a cero.

```
modelo_completo <- lm(FuelC ~ ., data=fuel2001)
summary(modelo_completo)
```

```
> summary(modelo_completo)
Call:
lm(formula = FuelC ~ Drivers + Income + Miles + MPC + Pop + Tax,
    data = fuel2001)
Residuals:
     Min       1Q   Median       3Q      Max
 -1480910  -158802    19267   174208  1090089

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -4.902e+05  8.199e+05  -0.598 0.552983
Drivers      6.368e-01  1.452e-01   4.386 7.09e-05 ***
Income       7.690e+00  1.632e+01   0.471 0.639793
Miles        5.850e+00  1.621e+00   3.608 0.000784 ***
MPC          4.562e+01  3.565e+01   1.280 0.207337
Pop         -1.945e-02  1.245e-01  -0.156 0.876586
Tax         -2.087e+04  1.324e+04  -1.576 0.122235
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
Residual standard error: 398400 on 44 degrees of freedom
Multiple R-squared:  0.9808,  Adjusted R-squared:  0.9782
F-statistic: 374.6 on 6 and 44 DF,  p-value: < 2.2e-16
```

En el modelo reducido solo tomamos las variables que deben ser diferentes de cero.

```
modelo_reducido <- lm(FuelC ~ Drivers + Miles + Pop, data=fuel2001)
anova(modelo_reducido)
```

```
> anova(modelo_reducido)
Analysis of Variance Table

Response: FuelC
          Df     Sum Sq    Mean Sq   F value    Pr(>F)
Drivers    1 3.5301e+14 3.5301e+14 2112.0851 < 2.2e-16 ***
Miles      1 2.8321e+12 2.8321e+12   16.9446 0.0001542 ***
Pop        1 8.7238e+10 8.7238e+10    0.5219 0.4735932
Residuals 47 7.8556e+12 1.6714e+11
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Comparamos ambos modelos.

```
anova(modelo_reducido, modelo_completo)
```

```
> anova(modelo_reducido, modelo_completo)
Analysis of Variance Table
Model 1: FuelC ~ Drivers + Miles + Pop
Model 2: FuelC ~ Drivers + Income + Miles + MPC + Pop + Tax
  Res.Df        RSS Df  Sum of Sq      F Pr(>F)
1     47 7.8556e+12
2     44 6.9843e+12  3 8.7128e+11 1.8296 0.1557
```

Vemos que el $p-$valor es mayor que $\alpha = 0.05$, por lo que se acepta la hipótesis planteada como cierta.

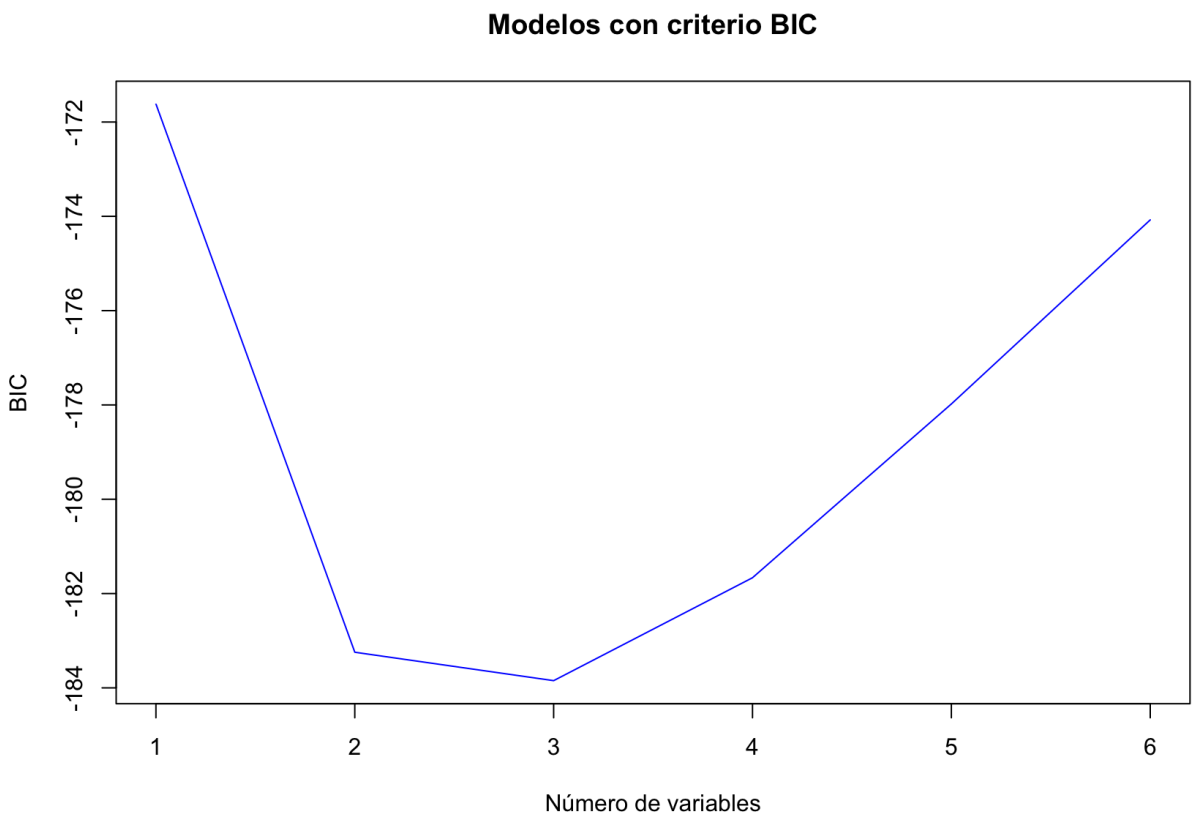
(c) De acuerdo con el método iterativo hacia adelante y el criterio BIC, ¿cuál es el modelo óptimo?

```
data <- data.frame(y=fuel2001$FuelC,
                   x1=fuel2001$Drivers,
                   x2=fuel2001$Income,
                   x3=fuel2001$Miles,
                   x4=fuel2001$MPC,
                   x5=fuel2001$Pop,
                   x6=fuel2001$Tax)

modelo_forward <- leaps::regsubsets(y ~ ., data=data, method='forward')
s_forward <- summary(modelo_forward)
s_forward
```

```
> s_forward
Subset selection object
Call: regsubsets.formula(y ~ ., data = datos, method = "forward")
6 Variables  (and intercept)
   Forced in Forced out
x1     FALSE      FALSE
x2     FALSE      FALSE
x3     FALSE      FALSE
x4     FALSE      FALSE
x5     FALSE      FALSE
x6     FALSE      FALSE
1 subsets of each size up to 6
Selection Algorithm: forward
         x1  x2  x3  x4  x5  x6
1  ( 1 ) "*" " " " " " " " " " "
2  ( 1 ) "*" " " "*" " " " " " "
3  ( 1 ) "*" " " "*" " " " " "*"
4  ( 1 ) "*" " " "*" "*" " " "*"
5  ( 1 ) "*" "*" "*" "*" " " "*"
6  ( 1 ) "*" "*" "*" "*" "*" "*"
```

```
plot(s_forward$bic ,xlab ='Número de variables',
     ylab="BIC", type="l", col='blue', main='Modelos con criterio BIC')
> s_forward$outmat[which.min(s_forward$bic), ]
 x1  x2  x3  x4  x5  x6
"*" " " "*" " " " " "*"
```

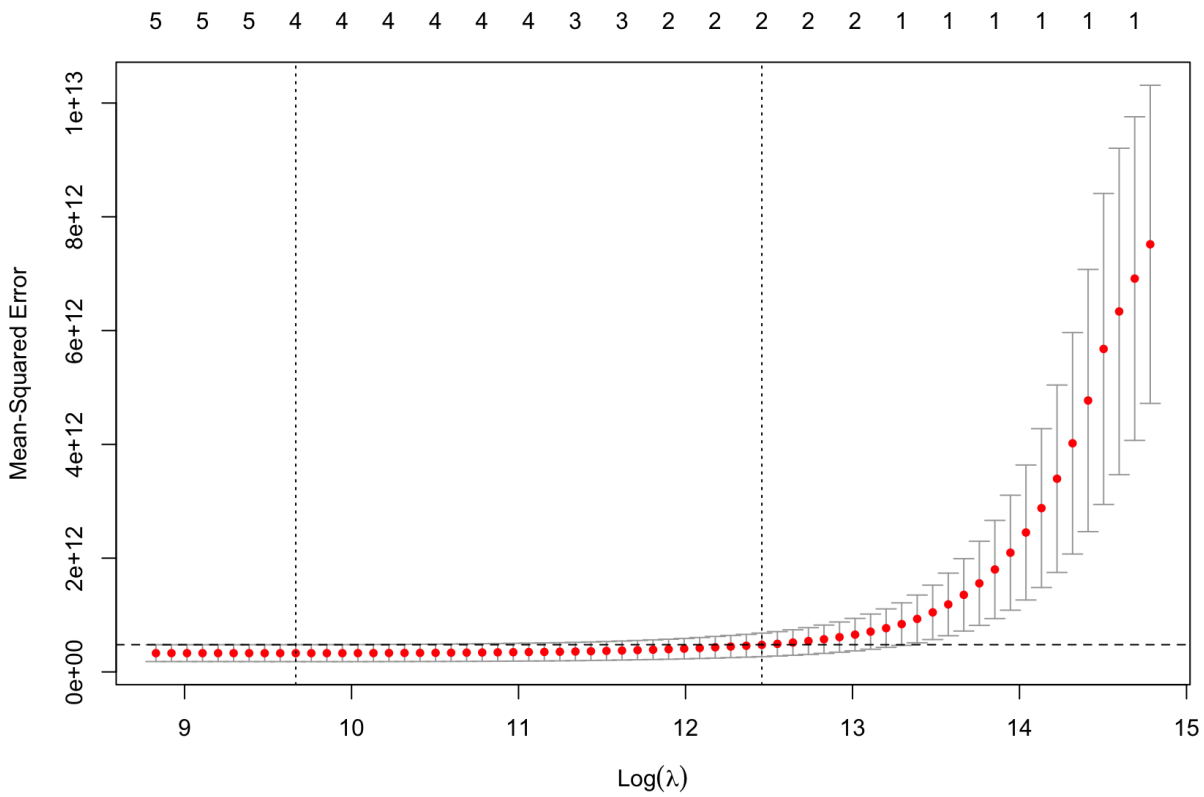


El modelo óptimo esta formado por 3 variables: $Drivers$, $Miles$ y Tax.

(d) Ajusta el modelo usando lasso, con el parámetro de regularización seleccionado mediante validación cruzada.

```
library(glmnet)
x <- as.matrix(datos[, -1])
y <- datos[, 1]

lasso <- cv.glmnet (x, y, alpha=1)# lasso -> alpha=1
lambda.lasso = lasso$lambda.1se
i <- which(min(lasso$cvm) == lasso$cvm)
abline(h = lasso$cvm[i] + lasso$cvsd[i], lty=2)
```




```
lambda.lasso = lasso$lambda.1se
final_lasso <- glmnet(x, y, alpha=1, lambda=lambda.lasso)
```

```
> final_lasso
Call:  glmnet(x = x, y = y, alpha = 1, lambda = lambda.lasso)

Df  %Dev Lambda
1  2 96.71 257000

> coef(final_lasso)
7 x 1 sparse Matrix of class "dgCMatrix"
                    s0
(Intercept) 1.528385e+05
x1          5.731942e-01
x2          .
x3          3.102360e+00
x4          .
x5          .
x6          .

> c <- lasso$beta[1:6, lasso$lambda == lambda.lasso]
> c[c != 0]
      x1       x3
0.573196 3.102269
```

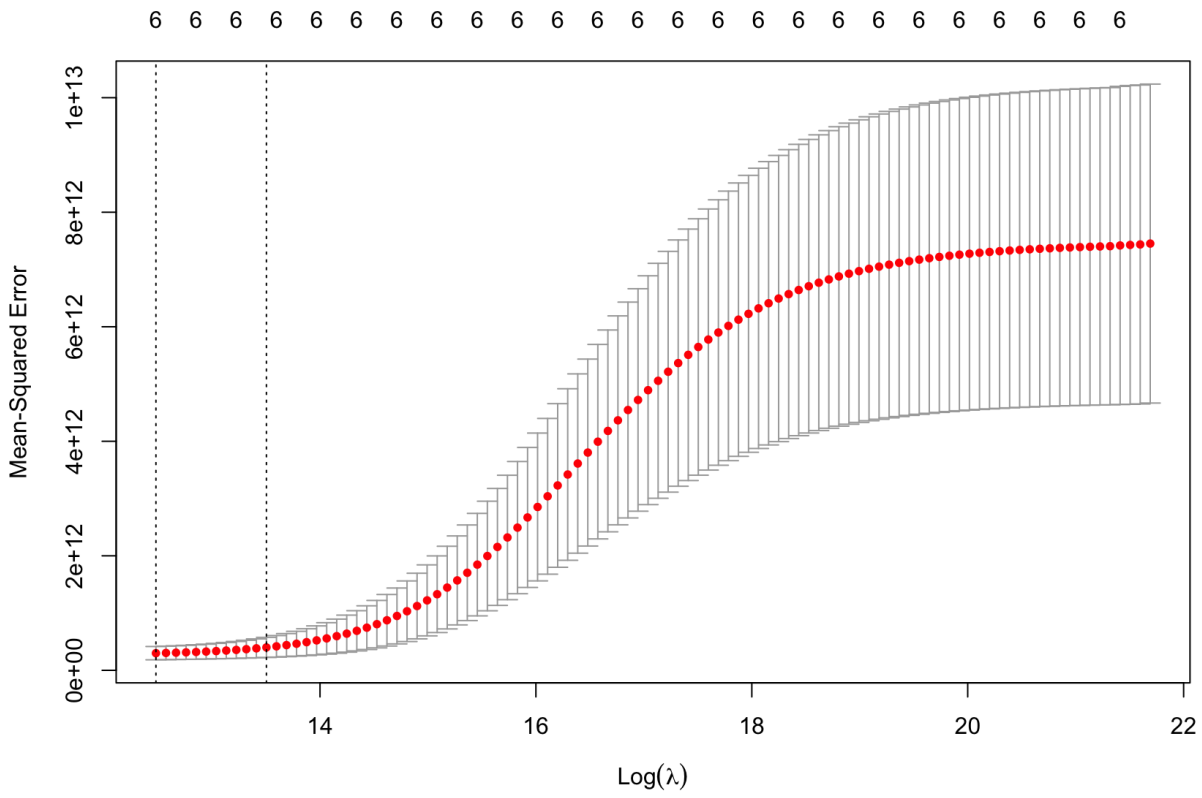
De esta forma mantenemos las variables $Drivers$ y $Miles$.

(e) Ajusta el modelo usando ridge, con el parámetro de regularización seleccionado mediante validación cruzada.

```
x <- as.matrix(datos[, -1])
y <- datos[, 1]

ridge <- cv.glmnet (x, y, alpha=0)  # lasso -> alpha=0
lambda.ridge = ridge$lambda.1se
```

```
i <- which(min(ridge$cvm) == ridge$cvm)
abline(h = ridge$cvm[i] + ridge$cvsd[i], lty=2)
```




```
lambda.ridge = ridge$lambda.1se
final_ridge <- glmnet(x, y, alpha=1, lambda=lambda.ridge)
```

```
> final_ridge
Call:  glmnet(x = x, y = y, alpha = 1, lambda = lambda.ridge)

Df  %Dev Lambda1
  1 89.52 732100

> coef(final_ridge)
7 x 1 sparse Matrix of class "dgCMatrix"
                    s0
(Intercept) 7.406584e+05
x1          4.805028e-01
x2          .
x3          .
x4          .
x5          .
x6          .
> c <- ridge$beta[1:6, ridge$lambda == lambda.ridge]
> c[c != 0]
          x1           x2           x3           x4           x5           x6
 2.590780e-01  1.909354e+01  8.966045e+00  2.056279e+00  2.083559e-01 -3.327424e+04
```

Este modelo es mucho menos restrictivo, obteniéndose todas las variables.